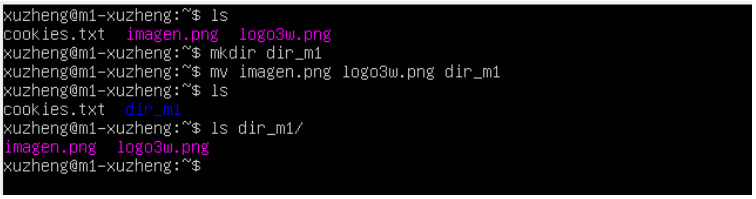
Práctica 2 SWAP

XuSheng Zheng

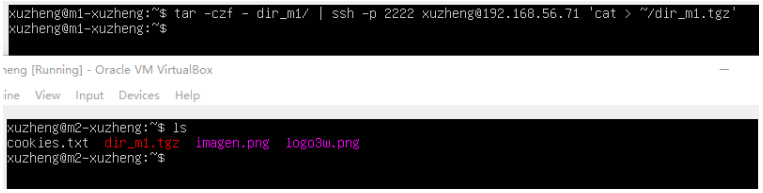
Índice

1. Copia de archivos

En primer lugar vamos a crear un directorio local en m1:



Vamos a enviar este directorio a m2 mediante **tar** y **ssh**:



Comprobamos descomprimiendo el archivo tar:

Esto también lo podemos llevar a cabo mediante scp. Puesto que habíamos configurado **SSH** en el puerto 2222, necesitamos indicarlo con el argumento **-P**:

2. RSync

En este apartado vamos a utilizar **rsync** para sincronizar ambas máquinas (de m1 a m2). En este caso, la herramienta ya está instalada por defecto. En primer lugar, necesitamos dar privilegio al usuario sobre la carpeta donde están los archivos del servidor web:

```
xuzheng@m1-xuzheng:~$ sudo chown xuzheng:xuzheng -R /var/www
[sudo] password for xuzheng:
xuzheng@m1-xuzheng:~$ _
```

Para probar, vamos a sincronizar la carpeta anterior:

```
xuzheng@m1-xuzheng:~$ rsync /var/www/ -avz -e 'ssh -p 2222' 192.168.56.71:/var/www/
sending incremental file list
rsync: chgrp "/var/www/." failed: Operation not permitted (1)
./
rsync: recv_generator: mkdir "/var/www/ejemplo" failed: Permission denied (13)
*** Skipping any contents from this failed directory ***
rsync: recv_generator: mkdir "/var/www/html" failed: Permission denied (13)
*** Skipping any contents from this failed directory ***
ejemplo/
html/

sent 242 bytes  received 379 bytes  414.00 bytes/sec
total size is 11,072  speedup is 17.83
rsync error: some files/attrs were not transferred (see previous errors) (code 23) at main.c(1207) [
sender=3.1.3]
xuzheng@m1-xuzheng:~$ _
```

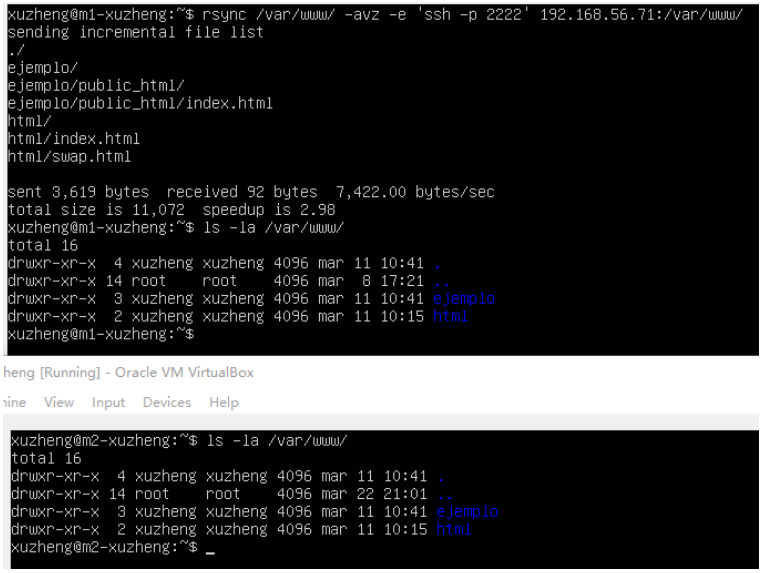
Podemos ver que en este caso no ha sido posible transferir la carpeta porque no tenemos permiso para escribir sobre ella. Si examinamos la carpeta */var* en m2:

```
xuzheng@m2-xuzheng:~$ ls -l /var/
total 48
drwxr-xr-x  2 root root   4096 mar 11 09:44 backups
drwxr-xr-x 13 root root   4096 mar  8 17:21 cache
drwxrwxrwt  2 root root   4096 ago 31  2022 crash
drwxr-xr-x 46 root root   4096 mar  8 17:21 lib
drwxrwsr-x  2 root staff  4096 abr 15  2020 local
lrwxrwxrwx  1 root root      9 ago 31  2022 lock -> /run/lock
drwxrwxr-x 11 root syslog 4096 mar 22 20:13 log
drwxrwsr-x  2 root mail   4096 ago 31  2022 mail
drwxr-xr-x  2 root root   4096 ago 31  2022 opt
lrwxrwxrwx  1 root root      4 ago 31  2022 run -> /run
drwxr-xr-x  5 root root   4096 ago 31  2022 snap
drwxr-xr-x  4 root root   4096 ago 31  2022 spool
drwxrwxrwt  7 root root   4096 mar 22 20:18 tmp
drwxr-xr-x  2 root root   4096 mar 22 21:01 www
xuzheng@m2-xuzheng:~$ _
```

nos damos cuenta de que sólo root tiene permiso para escribir sobre ella. Podemos otorgar el permiso al usuario en m2 también:

```
xuzheng@m2-xuzheng:~$ sudo chown xuzheng:xuzheng -R /var/www/
[sudo] password for xuzheng:
xuzheng@m2-xuzheng:~$
```

Y volvemos a intentar:



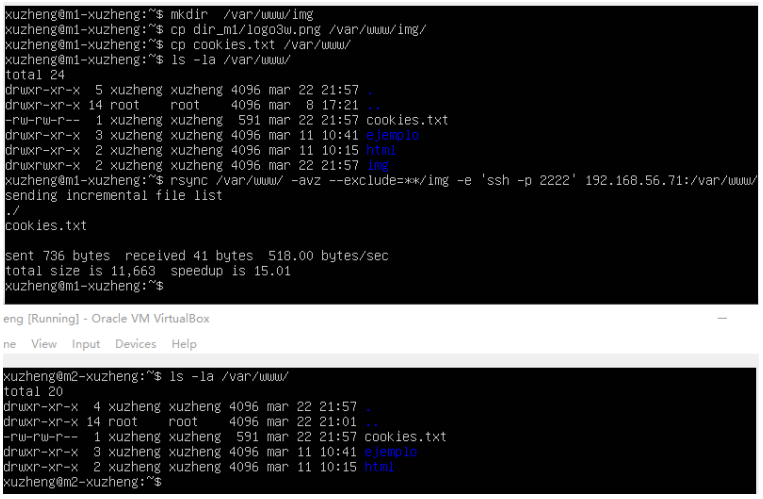
Podemos ver que ahora sí se ha sincronizado correctamente.

2.1. Opciones avanzadas

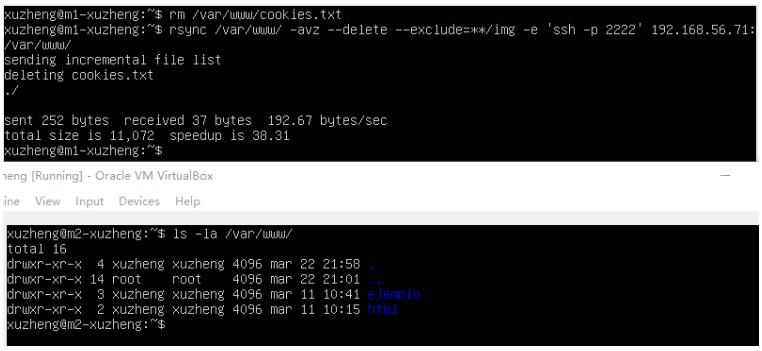
En la sincronización que hemos realizado anteriormente se ha utilizado los siguientes argumentos:

- **-a**: la transferencia se realiza en modo archivo, lo que asegura que los permisos, atributos, enlaces, etc se preserven y que la transferencia sea recursiva.
- **-v**: modo verbose, para dar más información.
- **-z**: comprime los archivos durante la transferencia.
- **-e**: determina el shell remoto a utilizar.

Como opciones avanzadas vamos a utilizar **–exclude** para evitar que se copie la carpeta */var/www/img* que vamos a crear, además vamos a copiar un archivo txt en */var/www* para eliminarlo después:



Podemos ver que ha excluido la carpeta */var/www/img* de la sincronización y sólo ha sincronizado el archivo txt. Ahora vamos a eliminar este archivo txt y probar con la opción –**delete**:

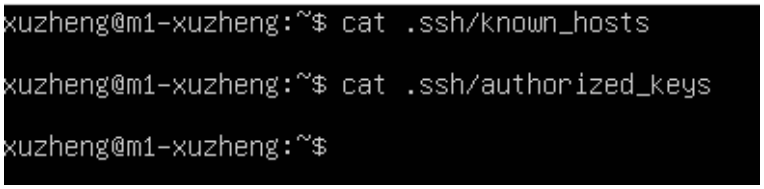


Podemos ver que como se ha eliminado el archivo en la máquina origen, también se ha borrado en la máquina destino.

3. SSH

En la práctica anterior, ya se configuró **ssh** para permitir el acceso sin contraseña utilizando **ssh-keygen** y **ssh-copy-id**. Para este apartado, vamos a borrar las claves y volver a generarlas:

Además necesitamos eliminar los contenidos de los archivos *.ssh/known_hosts* y *.ssh/authorized_keys*:



Ahora al intentar conectar nos pedirá la contraseña:

```
xuzheng@m1-xuzheng:~$ ssh -p 2222 xuzheng@192.168.56.70
The authenticity of host '[192.168.56.70]:2222 ([192.168.56.70]:2222)' can't be established.
ECDSA key fingerprint is SHA256:nHPleSchFq+XOPJFTLtyFvO/5VhaOkPEVOVinydcQL8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.56.70]:2222' (ECDSA) to the list of known hosts.
xuzheng@192.168.56.70's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-144-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of vie 24 mar 2023 15:36:46 UTC

  System load:  0.07              Processes:              122
  Usage of /:   42.2% of 8.02GB   Users logged in:        1
  Memory usage: 14%               IPv4 address for enp0s3: 10.0.2.15
  Swap usage:   0%                IPv4 address for enp0s8: 192.168.56.70


25 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
New release '22.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Fri Mar 24 15:30:21 2023 from 192.168.56.70
xuzheng@m1-xuzheng:~$ exit
logout
Connection to 192.168.56.70 closed.
xuzheng@m1-xuzheng:~$ _
```

Vamos a generar las nuevas claves de tipo **rsa** y 4096 bits en ambas máquinas:

```
xuzheng@m1-xuzheng:~$ ssh-keygen -b 4096 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/xuzheng/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/xuzheng/.ssh/id_rsa
Your public key has been saved in /home/xuzheng/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:6uBV8IQICfiAf3xAntGOnqTedWe1XyE+eOFc3VegHs8 xuzheng@m1-xuzheng
The key's randomart image is:
+---[RSA 4096]----+
| . o++o..     ...|
|. o..oo=     .  +|
| . +o.= .   oo.o+|
|  . ++.=   .=** o|
|   ...o S ..=*E .|
|   . . + . o.... |
|    o +       .  |
|   . +           |
|    . .          |
+----[SHA256]-----+
```

```
xuzheng@m2-xuzheng:~$ ssh-keygen -b 4096 -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/xuzheng/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/xuzheng/.ssh/id_rsa
Your public key has been saved in /home/xuzheng/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:qOhceI6pSO3CSWw0KkyzVIi7sjNpHQcwDc4wKoIBezk xuzheng@m2-xuzheng
The key's randomart image is:
+---[RSA 4096]----+
|*.o              |
|@*.o             |
|XXE.             |
|Xo+o   .         |
|+* .. . S        |
|.+++ o           |
|+O+.*            |
|*= B             |
|=== .            |
+----[SHA256]-----+
```

Ahora en lugar de utilizar **ssh-copy-id** vamos a utilizar **scp** para copiar la clave pública de una máquina al archivo *.ssh/authorized_keys* de otra:

```
xuzheng@m1-xuzheng:~$ scp -P 2222 .ssh/id_rsa.pub xuzheng@192.168.56.71:~/.ssh/authorized_keys
xuzheng@192.168.56.71's password:
id_rsa.pub                                    100%  744     1.2MB/s   00:00
xuzheng@m1-xuzheng:~$ _
```

uzheng [Running] - Oracle VM VirtualBox

achine View Input Devices Help

```
xuzheng@m2-xuzheng:~$ cat .ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDrxFdkQhDY7GgTosP9oIcBFhFEtGHatkWvbX5plnQY83qwbSrc/IhCmpajlYnv
ue4wV21Bm3VObO9fHd+4vQDSLqv3SXjPQN3YowhBOE4WMRNOC1JmUxfqNRCt9/IjNaUyATtgy8LIIh1fmZvtE2M5VD6TI2tv8bBK
w7u7b3MGMGgkq4OVKkGc5ZJSQAG2k9WNwnoUhQ9OB+T1OOg2p/GIsqRBmzCRWovGvTcZHqmrU1aAzVFuM7/+Z8OOlOTr/BXGxIDs
TF4PRWAOFgJT60fCfcJM2Rt6qlf89eK5DXqzUOhk2JTKmnwdIUqO6V8pKz2FjT/9lamm8rN91mGz+h/gfeWxKCUpI5V2UsvhdSei
nbZc5Mnu5XLh/IKXIJDo2FYKk+5VGuiBf5p6MCEQ2+YFSqL/JCVUp/OUfFP+Df2DuyT1FHbIEBjC1sZ+oUG2q93rWJavkNF5lCfw
NhN5gLHJMyFBtFjP+Oz7vo4WOs66OhuFcqIXZElRC8cWFOcG6Z21d5zvCKROdhEMr3ydgEOT/mlv8x6WdR1AxCze3FOEkKWMhl6f
ToIkz7XO5jhpJTLhwA/f2dKsRo1zHB6P4PEt3dFaJBSr76uSYWiIjLi2PmTti7Y+rqriqxEBKnw6K8cA/Hs4NfazHRdAVWBOaX2s
HmojRjqdRozx8kgEtbl0Mw== xuzheng@m1-xuzheng
xuzheng@m2-xuzheng:~$
```

Comprobamos conectando desde m1 a m2:

```
xuzheng@m1-xuzheng:~$ ssh -p 2222 xuzheng@192.168.56.71
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-144-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of vie 24 mar 2023 20:15:04 UTC

  System load:  0.0               Processes:              121
  Usage of /:   42.4% of 8.02GB   Users logged in:        1
  Memory usage: 14%               IPv4 address for enp0s3: 10.0.2.15
  Swap usage:   0%                IPv4 address for enp0s8: 192.168.56.71


25 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
New release '22.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Fri Mar 24 20:04:00 2023
xuzheng@m2-xuzheng:~$ exit
logout
Connection to 192.168.56.71 closed.
xuzheng@m1-xuzheng:~$
```

Como podemos observar, no nos ha pedido la contraseña.

4. Crontab

Para que se sincronice el directorio */var/www* entre las dos máquinas en cada hora, editamos el archivo */etc/crontab* como sigue:

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly
)
00 *    * * *   xuzheng rsync /var/www/ -avz --delete -e 'ssh -p 2222' 192.168.56.71:/var/www/
#
~
```

De forma que en el minuto 0 de cada hora se va a ejecutar **rsync**.

4.1. Opciones avanzadas

Aparte del * para indicar todos los valores, también podemos usar entre otras opciones: - para indicar el rango de tiempo y / para indicar el salto de tiempo. En el siguiente ejemplo, vamos a utilizar **crontab** para escribir la hora en un archivo en cada 90 minutos empezando por la media noche:

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly
)
00 *    * * *   xuzheng rsync /var/www/ -avz --delete -e 'ssh -p 2222' 192.168.56.71:/var/www
00 0-21/3 * * * root    date >> /home/xuzheng/hora_log.txt
30 1-22/3 * * * root    date >> /home/xuzheng/hora_log.txt
```

5. Bibliografía

- https://linux.die.net/man/1/scp
- https://linux.die.net/man/1/tar
- https://serverfault.com/questions/141773/what-is-archive-mode-in-rsync
- https://ss64.com/bash/rsync.html
- https://linux.die.net/man/1/rsync
- https://serverfault.com/questions/123629/run-task-every-90-minutes-with-cron